# 7 Application Environment

This section sets application environment standards that are not a part of the Oracle Designer repository. DCII operates on a three-tier architecture: database, application, and client.

The purpose of this chapter is to use consistent Unix directory structures for all DFAS Corporate Information Infrastructure (DCII) platforms, irrespective of server location. This document specifies how they will be used by the DCII applications. Also, a common environment file to specify all Unix application-level variables is established to provide ease of maintenance.

# 7.1 Application Directory Structures

## 7.1.1 Database Server Tier

### 7.1.1.1 Application Base Directory

ENV-01 A base directory structure will hold all the DCII application directories and subdirectories. This base directory may be located directly off the root, or under additional layers for Unix administration purposes. The base of the application directory structure will always be:

**/<mount_point>/dcii**

### 7.1.1.2 DCII Base Environment Variable

ENV-02 To facilitate easy referencing in navigating to or specifying such directory, a Unix environment variable to hold its value is defined. This will be **$DCII_BASE**. Using the first example above, this variable would be:

**$DCII_BASE=/<mount_point>/dcii**

### 7.1.1.3 Application Directories

ENV-03 Each application or module will have its own subdirectory. The subdirectory name will take the application prefix.

**/<mount_point>/dcii/<application_prefix>**

See Appendix B for approved application prefixes

**Example**: /fin090401/dcii/nsa

### 7.1.1.4 Release Directory Specification

ENV-04 The release number is created as a subdirectory to the application:

**/<mount_point>/dcii/<application prefix>/<release#>**

**Example**: /fin090401/dcii/nsa/0100

### 7.1.1.5 Event Directory Specification

ENV-05 As the application goes through the different phases, the subdirectory will be identified as follows:

**/<mount_point>/dcii/<application prefix>/<release#>/<event>**

where <event> is one of:

- dev# where # is available for multiple development phases

- sit
- fvt
- eit
- prod

**Example**: /fin090401/dcii/nsa/0203/sit

#### 7.1.1.6 Application Base Environment Variable

ENV-06 A Unix environment variable is created to point to each of the application subdirectories.

**$<application prefix>_BASE=/<mount_point>/dcii/<application prefix>/<release#>/<event>**

**Example**: $NSA_BASE=$DCII_BASE/nsa/0300/sit

#### 7.1.1.7 Application Component Subdirectories

ENV-07 For ease in identifying and locating what and where, the application installation and runtime components are grouped by their type and use. Figure 1.1 is a table outlining the class, group and the contents within that group.

| Subdirectory Class | Group Subdirectory | Contents |
|---|---|---|
| **Runtime Code** | …/bin | Pro*C programs, SQL*Loader control files, Unix shell scripts |
| | …/sql | SQL*Plus and PL/SQL interpretive code invoked at runtime |
| **Install-time Code** | …/admin | Top of all subdirectories for installation and administration code. |
| | …/admin/server | Install scripts to setup database objects such as tables, views, indexes, sequences, grants, synonyms, triggers, packaged procedures, and data conversion/migration scripts. |
| | …/admin/data | Data files used in migration, conversion and testing. |

*Fig. 1.1 Standard application components subdirectories*

**Example**:

To refer to the Unix shell scripts for DCII Release 0203 NSA, you would use:

dcii0203_sit.env

$NSA_BASE/bin/<unix_shell_script>

The following illustrates the standard database tier directory structure:

/fin090401/dcii

/fin090401/dcii/nsa
/fin090401/dcii/nsa/0203
/fin090401/dcii/nsa/0203/dev1
/fin090401/dcii/nsa/0203/dev1/bin
/fin090401/dcii/nsa/0203/dev1/sql
/fin090401/dcii/nsa/0203/dev1/admin
/fin090401/dcii/nsa/0203/dev1/admin/server
/fin090401/dcii/nsa/0203/dev1/admin/data
/fin090401/dcii/nsa/0203/sit
/fin090401/dcii/nsa/0203/sit/bin
/fin090401/dcii/nsa/0203/sit/sql
/fin090401/dcii/nsa/0203/sit/admin
/fin090401/dcii/nsa/0203/sit/admin/server
/fin090401/dcii/nsa/0203/sit/admin/data
/fin090401/dcii/nsa/0203/fvt
.
.
.
/fin090401/dcii/ceft/0203/dev1
.
.
.
/fin090401/dcii/ceft/0203/sit

ENV-08 References to these directories in modules must be established by referencing the appropriate environment variable set in the CSF system data table.

## 7.1.2 Application Server Tier

This section covers the configuration of the Application Server, or "mid-tier."

### 7.1.2.1 Universal Resource Locator (URL)

ENV-09 The URL for the production application server has been designated:

https://dcii.dfas.mil/

The URL for the Enterprise Integration test server has been designated:

https://eit-dcii.dfas.mil/

### 7.1.2.2 Application Base Directory

ENV-10 A base directory structure will hold all the DCII application directories and subdirectories. It will always be:

**<mount point>/dcii**

To facilitate easy referencing in navigating to or specifying such directory, a Unix environment variable to hold its value is defined. This will be **$WEBDCII.** For example, if the mount is the "fin090401" subdirectory of the root, this variable would be:

**$WEBDCII=/fin090401/dcii**

### 7.1.2.3 Application Directories

ENV-11 Each application or module will have its own subdirectory. The subdirectory name will take the application prefix.

**/<mount point>/dcii/<application_prefix>/<release#>/<event>/…**

See Appendix B for approved application prefixes

**Example**: The NSOA application will have the following:

- /fin090401/dcii/nsoa/0300/dev#/ where # is available for multiple development phases
- /fin090401/dcii/nsoa/0300/sit/
- /fin090401/dcii/nsoa/0300/fvt/
- /fin090401/dcii/nsoa/0300/eit/
- /fin090401/dcii/nsoa/0300/prod/

#### 7.1.2.4 Component Subdirectories

ENV-12 For ease in identifying and locating what and where, the application installation and runtime components are grouped by their type and use. Figure 1.2 is a table outlining the class, group and the contents within that group.

| Subdirectory Class | Group Subdirectory | Contents |
|---|---|---|
| Static and runtime HTML files | …/html | .html files (static and runtime) |
| Compiled | …/bin | .fmx, .mmx, and .rdf files (compiled) |
| Source | …/src | .fmb, .mmb, .rdf. (uncompiled) |

*Fig. 1.2 Standard application components subdirectories*

**Example**: For DCII Release 0300, the NSOA application will have the following:

- /fin090401/dcii/nsoa/0300/dev1/html
- /fin090401/dcii/nsoa/0300/dev1/bin
- /fin090401/dcii/nsoa/0300/dev1/src

ENV-13 All icons used in forms will be converted to GIF files to be used over the web. The Common Service Functions (CSF) objects (icons, libraries) have been loaded and must be referenced. This will be implemented using the .profile environment settings. These are located in the following directories:

- /<mount-point>/dcii/csf/<release_number>/<event>/lib for libraries
- /<mount-point>/dcii/csf/<release_number>/<event>/img for icons

**Example**: For DCII Release 0300, the NSOA application will have the following:

- /fin090401/dcii/csf/0300/dev1/lib
- /fin090401/dcii/csf/0300/dev1/img

Developers must first compile all Designer Forms and Reports within the PC-based Designer environment prior to porting to the Unix platform. This must be done using the DCII-SHARED templates and libraries. Only .fmb and .rdf should be ported to the Unix platform, as the executables must be re-compiled on the target platform.

#### 7.1.2.5 Web Server Forms

ENV-14 The standard URL for web server forms will be named using the convention:

https://dcii.dfas.mil/dciiplsql/<plsql-form-name>

PLSQL forms will access static HTML and image content by referring to the virtual directories using the convention

/dcii-html

/dcii-img

### 7.1.3 FTP Directory Structure

ENV-15 All applications that use File Transfer Protocol (FTP) run through the File Inventory Control Subsystem (FICS). The subdirectory structure for FICS is:

/<mount_point>/dcii/<application_prefix>_<fics_project>/<release_number>/<event>/

| Variable | Directory |
|---|---|
| $SRCDIR | …inbox |
| $PGMDIR | …/bin |
| $WRKDIR | …/data_in |
| | …/data_out |
| | …/archv |
| | …/unknown |
| $LOGDIR | …/log |

**Example**:

- /fin090401/dcii/nsa_caps/0203/sit/bin
- /fin090401/dcii/ceft_caps/0203/sit/bin
- /fin090401/dcii/get_adm/0203/sit/bin

ENV-16 Use the environment variables indicated for all references to the FICS directory structure in shell scripts. Do not use hard coded references.

## 7.2 Application Environment Setup

### 7.2.1 Environment Variables File

ENV-17 All the DCII environment variables will be in one environment file, which can then be referenced by programs that would use these variables. This provides flexibility (changing the values without changing the programs) and ease of maintenance (change is done only in one location).

ENV-18 The shell script defining the environment variables will be named using the convention:

**/etc/dcii/dcii<release><suffix>.env**

Where:

<release> is the release number in the form of "xxxx".

<suffix> is an indicator of the environment to be established when the server supports multiple environments. The following suffixes are valid:

- Development _dev#
- System Integration Test _sit
- Functional Validation Test _fvt
- Enterprise Tests (all) <none>
- Production <none>

**Example**:

| | |
|---|---|
| dcii0300_dev.env | Development for DCII Release 0300. |
| dcii0203.env | EIT and Production for DCII Release 0203. |

ENV-19 Applications should reference their directory trees indirectly, using the environment variable representing the home directory of the application.

### 7.2.2 Sample Environment file

#### 7.2.2.1 SIT Environment - dcii0203_sit.env

```
#--   file:     dcii0203_sit.env
#--   author:   R. Strommen 8/2000
#--   Overview: Environment variables for DCII Release 0203 SIT
#--   Change History (when, who, what)
#--   8/00 – RLS – Create script
#--
DCII_BASE=/fin090401/dcii
export DCII_BASE
CEFT_BASE=$DCII_BASE/ceft/0203/sit
export CEFT_BASE
DCW_BASE=$DCII_BASE/dcw/0203/sit
export DCW_BASE
DDRS_BASE=$DCII_BASE/ddrs/0203/sit
export DDRS_BASE
DSDS_BASE=$DCII_BASE/dsds/0203/sit
export DSDS_BASE
FICS_BASE=$DCII_BASE/fics/0203/sit
export FICS_BASE
GET_BASE=$DCII_BASE/get/0203/sit
export GET_BASE
NSA_BASE=$DCII_BASE/nsa/0203/sit
```

```
export NSA_BASE
NSOA_BASE=$DCII_BASE/nsoa/0203/sit
export NSOA_BASE
SGL_BASE=$DCII_BASE/sgl/0203/sit
export SGL_BASE
```

#### 7.2.2.2 Production environment - dcii0203.env

```
#--   file:     dcii0203.env
#--   author:   R. Strommen 8/2000
#--   Overview: Environment variables for DCII Release 0203
#--   Change History (when, who, what)
#--   8/00 – RLS – Create script
#--
DCII_BASE=/fin090401/dcii
export DCII_BASE
CEFT_BASE=$DCII_BASE/ceft/0203/prod
export CEFT_BASE
DCW_BASE=$DCII_BASE/dcw/0203/prod
export DCW_BASE
DDRS_BASE=$DCII_BASE/ddrs/0203/prod
export DDRS_BASE
DSDS_BASE=$DCII_BASE/dsds/0203/prod
export DSDS_BASE
FICS_BASE=$DCII_BASE/fics/0203/prod
export FICS_BASE
GET_BASE=$DCII_BASE/get/0203/prod
export GET_BASE
NSA_BASE=$DCII_BASE/nsa/0203/prod
export NSA_BASE
NSOA_BASE=$DCII_BASE/nsoa/0203/prod
export NSOA_BASE
SGL_BASE=$DCII_BASE/sgl/0203/prod
export SGL_BASE
```

## 7.3 Interface Data File Formats

This standard applies to all interfaces created directly for DCII applications. Where legacy and external systems have existing formats, these will be accepted.

ENV-20 The data file format will be comma delimited for all data loads. The delimiter may be any special character not typically in the raw data. All characters in the interface file must be convertible to ASCII.

ENV-21 Filenames must be unique. This will avoid files from being overwritten during FTP.

ENV-22 Filenames must not exceed 40 characters. The filename is loaded as part of the NSA header record. The filename column supports up to 40 characters.

ENV-23 The header record must be the first line in the file.

ENV-24 The header record should have SOURCE SYSTEM NAME and DATE and TIME created. This is a unique id defined to prevent loading the same file twice.

ENV-25 Site value should exist in either the filename or the header record. Only files whose source system and site exist in a predefined source system and site table are processed.

ENV-26 The header record should have RECORD COUNT and TOTAL AMOUNT. Each file is validated such that the both the record count and total amount in the header record should match with both the number of records and sum of the amount in the detail records. This will ascertain no data lost between the file creation time until the time it is processed in the Non-Standard Area (NSA).

Validation of the record count and total amount is done using the awk command of Unix. If there are non-numeric characters in the detail amount field, awk will take only the numeric values before the occurrence of the non-numeric character. If the non-numeric character occurs first, awk will interpret the value as 0. (i.e. 0050A is interpreted as 50; A002 is interpreted as 0) There is a small, but acceptable, risk of still getting a match on the amount.

ENV-27 There should not be any blank detail records. If there are no mandatory columns defined for the corresponding detail table, a blank record will be loaded. To prevent a blank record from being loaded, either define a mandatory column, or eliminate blank detail records from the datafile.

ENV-28 There should be no null characters in the file. This would cause the rest of the line after the null to be dropped during FTP.

ENV-29 Numeric and Date formats should be consistent. Only one format can be specified in a SQL*Loader control file.

ENV-30 There must not be a space between the sign character and the value in the detail record. This will be interpreted incorrectly by SQL*Loader. Use a floating sign character for the value.

ENV-31 All blank fields will be considered as null values when loaded.

## 7.4 Database Configuration

### 7.4.1 Database Environments

There are six database environments supporting the DCII:

- Island Development
- Integrated Development
- System Integration Test
- Functional Validation Test
- Enterprise Integration Test / Enterprise Performance Test / Enterprise Acceptance Test
- Operational Test and Evaluation / Production

Island development is where the application code is developed and unit tested.

Integrated development is where code from several applications for a release first comes together.

### 7.4.2 Key Database Parameters

Database parameters are set based on the hardware capacity and the size of the database. All test environments (PROD, FVT, EIT, EPT, EAT) and production should be equivalent sized platforms and the database parameters should be set to the same value.

ENV-32 All database environments will use the same operating system and software versions as the production system they are targeted for.

ENV-33 Initialization Parameter Values are:

| Parameter | Oracle Default | Test Systems Dev, SIT, FVT | Production EIT, EPT, Prod |
|---|---|---|---|
| DB_BLOCK_SIZE | 8192 | TBD | TBD |
| DML_LOCKS | 500 | TBD | TBD |
| ENQUEUE_RESOURCES | 5000 | TBD | TBD |
| NLS_DATE_FORMAT | DD-MON-RR | TBD | TBD |
| NLS_LANGUAGE | AMERICAN | TBD | TBD |
| NLS_NUMERIC_CHARACTERS | “.” | TBD | TBD |
| NLS_SORT | BINARY | TBD | TBD |
| NLS_TERRITORY | AMERICA | TBD | TBD |
| OPEN_CURSORS | 255 | TBD | TBD |
| OPTIMIZER_MODE | RULE | TBD | TBD |
| _OPTIMIZER_UNDO_CHANGES | TRUE | TBD | TBD |
| PROCESSES | 75 | TBD | TBD |
| ROW_LOCKING | ALWAYS | TBD | TBD |
| SHARED_POOL_SIZE | 31,457,280 | TBD | TBD |
| SORT_AREA_SIZE | 256,000 | TBD | TBD |
| GLOBAL_NAMES | | TRUE | TRUE |
| OPEN_LINKS | | 4 | 4 |
| DISTRIBUTED_LOCK_TIMEOUT | | 300 | 300 |
| DISTRIBUTED_TRANSACTIONS | | 5 | 5 |
| JOB_QUEUE_PROCESSES | | 10 | 10 |
| PARALLEL_MAX_SERVERS | | 10 | 10 |
| PARALLEL_MIN_SERVERS | | 2 | 2 |

### 7.4.3 Key Account Security Parameters

The following account security parameters are set in an application profile. These security parameters make the Oracle 8i infrastructure compliant with the Department of Defense's Minimum Security Requirements.

ENV-34 Account profiles will be named as follows:

- Users DCIIUSER
- Test DCIITEST (Only for tester accounts testing security)
- Applications DCIIAPPS

ENV-35 Profile Account Security Parameter Values are:

| **Profile Parameter** | **DCIITEST** | **DCIIUSER** |
|---|---|---|
| FAILED_LOGIN_ATTEMPTS (#) | 3 | 3 |
| PASSWORD_LOCK_TIME (days) | 30/1440 | 30/1440 |
| PASSWORD_LIFE_TIME (days) | 3 | 81 |
| PASSWORD_GRACE_TIME (days) | 1 | 9 |
| PASSWORD_REUSE_MAX (days) | 7 | 730 |
| VERIFY_FUNCTION<br>Minimum Length | 8 | 8 |

ENV-36 Whenever an account is created, or the password is reset by the Help Desk, then the password must be set as expired. This will require that the user changes the password upon the initial use, and therefore only the user will know the password. The password can be set as expired by using the command:

ALTER USER <usercode> PASSWORD EXPIRE;